# Privacy-Preserving Population-Enhanced Biometric Key Generation from Free-Text Keystroke Dynamics*

Jaroslav Šeděnka†
New York Inst. of Technology
jsedenka@nyit.edu

Kiran S. Balagani
New York Inst. of Technology
kbalagan@nyit.edu

Vir Phoha
Louisiana Tech University
phoha@latech.edu

Paolo Gasti
New York Inst. of Technology
pgasti@nyit.edu

## Abstract

*Biometric key generation techniques are used to reliably generate cryptographic material from biometric signals. Existing constructions require users to perform a particular activity (e.g., type or say a password, or provide a handwritten signature), and are therefore not suitable for generating keys continuously. In this paper we present a new technique for biometric key generation from free-text keystroke dynamics. This is the first technique suitable for continuous key generation. Our approach is based on a scaled parity code for key generation (and subsequent key reconstruction), and can be augmented with the use of population data to improve security and reduce key reconstruction error. In particular, we rely on linear discriminant analysis (LDA) to obtain a better representation of discriminable biometric signals.*

*To update the LDA matrix without disclosing user's biometric information, we design a provably secure privacy-preserving protocol (PP-LDA) based on homomorphic encryption. Our biometric key generation with PP-LDA was evaluated on a dataset of 486 users. We report equal error rate around 5% when using LDA, and below 7% without LDA.*

## 1. Introduction

Biometric Key Generation (BKG) harnesses biometric signals to protect cryptographic keys against unauthorized access. Freshly-generated keys are *committed* using biometric information; subsequently, the same biometric signal (or a "close enough" signal) is used to reconstruct (i.e., *decommit*) a key.

BKG offers unique and appealing features. Unlike passwords, biometric information (and the corresponding key) is tied to a particular user, and as such cannot be easily disclosed or stolen (e.g., via shoulder-surfing). Easy-to-remember passwords provide only marginal security, while strong passwords are difficult to remember. Ideally, by relying on high-entropy and consistent biometric signals, BKG is a good candidate for replacing password-based techniques, because it offers a good tradeoff between usability and security.

Although originally conceived for physical biometrics [11], such as fingerprints and iris, BKG techniques have recently been applied to *behavioral biometrics*. These techniques include key generation using voice [13], handwritten signatures [7, 9] and keystroke dynamics [14]. Due to the inherent variability of behavioral signals, current approaches require users to perform a specific activity while these signals are collected. For example, techniques based on voice recognition require users to pronounce the same sentence (i.e., a passphrase) for both committing and decommitting a key; BKG based on keystroke dynamics *augments* password-based systems by introducing additional entropy while the user is typing her password.

This work is the first to introduce BKG on *free-text* input. (In free-text setting, users are allowed to type or say any text.) Free-text BKG allows periodic key generation using behavioral data collected continuously, since collection of biometric signals does not interfere with regular user activity. Therefore, free-text based systems can capture biometric signals over a long period of time, providing better accuracy and security.

**Contributions.** We propose a novel BKG technique that builds on the fuzzy commitment schemes of Juels *et al*. [11]. Our work represents a further step towards a formalization of security of BKG techniques. We define new and more realistic requirements for biometric signals. In particular, while fuzzy commitments of Juels *et al*. are secure under the unrealistic assumption that all biometric features are uni-

*This work was supported in part by DARPA Active Authentication grants FA8750-12-2-0201 and FA8750-13-2-0274 and NYIT ISRC 2012 and 2013 grants. The views, findings, recommendations, and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the sponsoring agencies or the U.S. Government.

†J. Šeděnka (sedenka@mail.muni.cz) is a student of Faculty of Science, Masaryk University, Czech Republic. This work was done while visiting the New York Institute of Technology.

formly distributed, our commitments are provably secure without any assumption on the distribution of the biometric features. Instead, we assume that biometric signals are an instantiation of an *unpredictable function* [15], which is a well-understood cryptographic tool.

Furthermore, we extend the error-correcting code to work on arbitrary biometrics. We then improve commitment/decommitment performance by relying on population data. We evaluate the feasibility of using Linear Discriminant Analysis [8] (LDA) for improving the EER of our technique. Since LDA requires data from multiple users, we design a privacy-preserving protocol (PP-LDA) that allows users to compute LDA parameters without disclosing their biometric signals. The protocol involves two untrusted parties: an *enrollment server* (ES) and a *matrix publisher* (MP). To our knowledge, this work is the first to use LDA (and, consequently, PP-LDA) for the purpose of biometric key generation.

By using keystroke and digraph latency information, we were able to achieve 5.5% false accept rate (FAR) and 3.6% false reject rate (FRR).

**Organization.** The rest of the paper is organized as follows. In Section 2 we overview the tools and the security model used in this paper. Section 3 introduces our modification of Fuzzy Commitment, and presents our PP-LDA protocol. In Section 4 we analyze the security of our BKG technique and of PP-LDA. Experimental evaluation is presented in Section 5. We summarize related work in Section 6 and conclude in Section 7.

## 2. Background

**LDA.** Linear Discriminant Analysis [8] (LDA) is a well known supervised feature extraction method. The goal of LDA is to derive a new (and possibly compact) set of features from the original feature set, such that the new set provides increased class-discriminability. Formally, LDA finds direction vectors for projections that maximize linear separability between classes ('users' in our case). Let $v$ denote the number of users, $X_i$ be the $m_i \times n$ matrix of data samples containing $m_i$ $n-$dimensional training samples of user $i$. The mean of $X_i$ is denoted by the row vector $\mu_i$ and $\mu$ is the mean of all $\mu_i$-s. Let $C_m$ denote the $m \times m$ centering matrix.

LDA finds the transformation matrix $W = [w_1; \ldots; w_{n-1}]$ that maximizes the objective function $J(W) = \frac{|W^T S_b W|}{|W^T S_w W|}$, where the *scatter between* term '$S_b$' is calculated as $\sum_{i=1}^{v}(\mu_i - \mu)^T(\mu_i - \mu)$ and the *scatter within* term '$S_w$' is calculated as $\sum_{i=1}^{v} X_i^T C_{m_i} X_i$. It can be easily shown that the transformation matrix $W$ is the solution of the generalized eigenvalue problem $S_B W = \Lambda S_W W$. After the transformation matrix is obtained, the new $(n-1)$ features are calculated as $XW$.

**Fuzzy Commitments and BKG.** Fuzzy commitments [11] use error correcting codes to construct commitments from noisy information, e.g., biometric signals. Features are extracted from raw signals (e.g., minutiae from fingerprint images); then, each feature is represented using a single bit, therefore constructing an $n$-bit vector (where $n$ is the number of features) for each sample. Let $C \subseteq \{0,1\}^n$ be a group error-correcting code. To commit a codeword $c \in C$ using biometric $x = (x_1, \ldots, x_n)$, the user computes commitment $(\mathrm{H}(c), \delta = x \oplus c)$; the biometric key is computed as $k = \mathrm{H}'(c)$, where H and H′ are two collision resistant hash functions.

To decommit the biometric key using biometric sample $y = (y_1, \ldots, y_n)$, the user computes codeword $c' = \mathsf{decode}(y \oplus \delta)$. If $\mathrm{H}(c') = \mathrm{H}(c)$, then the biometric key is computed as $k = \mathrm{H}'(c')$. Otherwise, no information about the key is revealed. Biometric vector $y$ decommits $c$ iff the error vector $e = x - y$ decodes to the zero codeword.

Juels *et al.* [11] prove that the adversary cannot reconstruct $c$ from a commitment under the assumption that H and H′ are collision-resistant functions and that $x$ is uniformly distributed in $\{0,1\}^n$.

**Unpredictable functions.** In contrast with the technique in [11], we model biometric features as unpredictable functions [15]. This captures the idea that a user's biometric is *difficult to guess*. Informally, an unpredictable function $f(\cdot)$ is a function for which no efficient adversary can predict $f(x^*)$ given $f(x_i)$ for various $x_i \neq x^*$. Formally:

**Definition 1.** *A function family $(\mathcal{C}, D, R, F)$ for $\{f_c(\cdot) : D \to R\}_{c \leftarrow \mathcal{C}}$ is unpredictable if for any efficient algorithm $\mathcal{A}$ and auxiliary information $z$ we have:*

$$Pr[(x^*, f_c(x^*) \leftarrow \mathcal{A}^{f_c(\cdot)}(z) \text{ and } x^* \notin Q] \leq \mathrm{negl}(\kappa)$$

*where $Q$ is the set of queries from $\mathcal{A}$, $\kappa$ is the security parameter and $\mathrm{negl}(\cdot)$ is a negligible function.*

**Homomorphic Encryption.** Our PP-LDA construction uses a semantically secure (public key) additively homomorphic encryption scheme. Let $[\![m]\!]$ indicate the encryption of message $m$ using a homomorphic encryption scheme. (We omit public keys in our notation, since there is a single public/private keypair generated by MP.) We have that $[\![m_1]\!] \cdot [\![m_2]\!] = [\![m_1 + m_2]\!]$, which also implies that $[\![m]\!]^a = [\![m \cdot a]\!]$. While any encryption scheme with the above properties suffices for the purposes of this work, the construction due to Damgård *et al.* [6, 5] (DGK hereafter) is of particular interest here because it is fast and produces small ciphertexts.

Fully-homomorphic encryption (FHE) can also be used to instantiate our PP-LDA protocol. However, due to the severe performance penalty associated with FHE, we design a protocols that requires only additively homomorphic encryption.

**Security Model and Definitions.** Our protocols are secure in the presence of semi-honest (also known as honest-but-curious or passive) participants. In this model, while participants follow prescribed protocol behavior, they might try to learn additional information beyond that obtained during normal protocol execution.

We use the term *adversary* to refer to insiders, i.e., protocol participants. This includes the case when one of the parties is compromised. Outside adversaries, e.g., those who can eavesdrop on the communication channel, are not considered since their actions can be mitigated via standard network security techniques (e.g., by performing all communication over SSL).

## 3. Our Techniques

In this section we introduce our BKG construction. To generate a cryptographic key, the user selects a random codeword $c$ of length $n$ from a custom error-correcting code $C$, and uses it to derive a cryptographic key as $k = \mathrm{PRF}_c(z)$, where PRF is a pseudorandom function and $z \neq 0$ is either a system-wide public constant or a user-provided pin/password for added security. (In our security analysis we assume that the adversary knows $z$) $c$ is then protected using the user's biometrics as follows. After collecting keystroke data, we extract $n$ keystroke and digraph features $x = (x_1, \ldots, x_n)$, which are then discretized and scaled by their standard deviation. The user then computes $\delta = (x - c) = (x_1 - c_1, \ldots, x_n - c_n)$ and publishes commitment $(\mathrm{PRF}_c(0), \delta)$.

The user can reconstruct the cryptographic key given public parameters $(\mathrm{PRF}_c(0), \delta)$, a biometric signals and possibly a pin/password $z$ as follows. The user extract keystroke/digraph features from the sample; then she constructs vector $y = (y_1, \ldots, y_n)$. Finally, she computes $c' = \mathsf{decode}(y - \delta)$. If $\mathrm{PRF}_{c'}(0) = \mathrm{PRF}_c(0)$, then $k = \mathrm{PRF}_{c'}(z)$ is the correct key with overwhelming probability. In the following, we provide further details on our construction.

**Our Construction.** In our scheme, each feature is discretized and mapped to the range $[0, 2^d - 1]$. In other words, codeword symbols are elements of $\mathbb{Z}_{2^d}$. Discretization is performed as:

$$\mathsf{discretize}_{d,\mathsf{F}}(x_j) = \left\lfloor (2^d - 1)\left(\frac{x_j - \mathsf{min}_\mathsf{F}}{\mathsf{max}_\mathsf{F} - \mathsf{min}_\mathsf{F}}\right)\right\rfloor$$

where $\mathsf{F}$ is the feature being discretized, $x_j$ is an instance in $\mathsf{F}$, $\mathsf{min}_\mathsf{F}$ is the minimum value of $\mathsf{F}$, and $\mathsf{max}_\mathsf{F}$ is the maximum value of $\mathsf{F}$. The $d$ parameter controls the number of cells a feature is discretized into. Therefore, higher the $d$, the lower the potential loss of information due to discretization. When $x_j > \mathsf{max}_\mathsf{F}$ or $x_j < \mathsf{min}_\mathsf{F}$, we set $\mathsf{discretize}_{d,F}(x_j)$ to $2^d - 1$ and 0, respectively.

$c =$ | **1101**000 | **01010**00 | **01**00000 | **100**$p$000 |
$c_1$ $c_2$ $c_3$ $c_4$

Figure 1. Example of a SPC codeword. Each codeword symbol $c_i$ for $i < n$ ends with $l_i$ zero bits. (We write the least significant $l_i$ bits of each symbol using non-bold typeface.) The last symbol ends with $l_n - 1$ zero bits, preceded by one parity bit, denoted with $p$ in the figure.

Distance between two codewords is not defined via the usual Hamming distance. In fact, Hamming distance captures well the "similarity" between a bit string and its perturbed version when all bits in the string have the same probability of being affected by an error. In our setting this is not the case, because the least significant bits of each feature instance have higher probability of being altered.

Therefore, we instantiate fuzzy commitments using a custom-designed error-correcting code inspired by codes in the Lee metric [12]. Let us define Lee weight and Lee distance as follows:

**Definition 2** (Lee weight)**.** *The* Lee weight *of element $x \in \mathbb{Z}_{2^d}$ is defined as $w_L(x) = abs(x')$, such that $x' \equiv x \bmod 2^d$ and $-2^{d-1} < x' \leq 2^{d-1}$. The* Lee weight *of vector $x = (x_1, \ldots, x_n) \in (\mathbb{Z}_{2^d})^n$ is defined as the sum of Lee weights of its elements, i.e., $w_L(x) = \sum_{i=1}^{n} w_L(x_i)$.*

**Definition 3** (Lee distance)**.** *The* Lee distance *of vectors $x, y \in \mathbb{Z}_{2^d}$ is the Lee weight of their difference, i.e., $d_L(x, y) = w_L(x - y)$.*

We consider each feature vector a (possibly perturbed) codeword of a code in Lee metric, and we use the *Lee weight* as a metric for distance between feature vectors. We refer to individual elements in a codeword (i.e., individual features) as *symbols*. Features are scaled by the standard deviation and discretized as $s'_i = \mathsf{discretize}_{d,\mathsf{F}}(\sigma_i \cdot \kappa)$, where $\sigma_i$ is the standard deviation of feature $\mathsf{F}$. Then $s'_i$ is mapped to the closest power of two, which we indicate as $s_i$. The $l_i$ least significant bits of $c$ are zero in all codewords. (The last symbol has $l_n - 1$ bits set to zero since the least significant bit is used for parity.)

Existing codes in the Lee metric only guarantee correct decoding when the Lee weight of the error is a small multiple of the number of codeword symbols [23, 18]. However, in our setting the number of codeword symbols is relatively small (i.e., between 20 and 100), while the domain for each feature is large (integers between 0 and $2^{24} - 1$ in our experiments). Therefore, we use group error-correcting code, which we refer to as a *scaled parity code* (SPC). Let $n$ denote the number of features, $d$ be the discretization parameter, $\kappa$ the security parameter. A vector $c = (c_1, \ldots, c_n) \in (\mathbb{Z}_{2^d})^n$ is a SPC codeword iff it satisfies the following two conditions: (a) for all $i \in \{1, \ldots, n\}$, $s_i$ divides $c_i$; and (b) $\sum_{i=1}^{n} c_i / s_i \equiv 0 \mod 2$ (i.e., *parity condition*). Figure 1 shows the structure of a sample codeword.

If the parity condition is not met during decoding, we select $c'_k$ such that $|y_k - x_k|$ is assumed to be largest (after normalization) among all $|y_i - x_i|$. The parity is corrected by adding (subtracting) $s_k$ to $c'_k$ if $(y_k - \delta) - c'_k$ is positive (negative, respectively).

Our SPC is designed to guarantee that only feature vectors "close" to the user's template decommit to the correct codeword. When the number of codeword symbols is either one or two, the SPC algorithm decodes vectors to the closest codeword in the Lee metric.

**Theorem 1.** *Let $n \in \mathbb{N}$ be the number of features, $d \in \mathbb{N}$ the discretization parameter, $C \subset \mathbb{Z}_{2^d}^n$ a scaled parity code with scaling $s_1, ..., s_n$, $c \in C$ be a codeword and $\gamma = c + \epsilon$ where $\epsilon \in \mathbb{Z}_{2^d}^n$ is the error. If the sum of the two biggest relative errors is smaller than one, Algorithm 1 decodes $\gamma$ to $c$.*

(Due to space constraints, proof of Theorem 1 is omitted, and is available in the full version of the paper [**?**].)

**Algorithm 1** DECOMMIT CODEWORDS IN SPC

0: on input $\gamma = (\gamma_1, \ldots, \gamma_n)$,
scaling factors $s = (s_1, ..., s_n)$:
1: **for** each feature $i$ **do**
2: $e_i = \gamma_i \bmod s_i$ `// e_i is error on feature i`
3: **if** $e_i/s_i > 1/2$ **then**
4: $e_i = e_i - s_i$
5: **end if**
6: **end for**
7: $c' = \gamma - e$ `//subtract error`
8: $p = \sum_{i=1}^{n}(c'_i/s_i)$ `//check parity`
9: **if** $p \equiv 1 \pmod 2$ **then**
10: $k = argmax_i(|e_i/s_i|)$
`//feature index with max. relative error`
11: $c'_k = c'_k + sign(e_k)s_k$
12: **end if**
13: **return** $c' = (c_1, \ldots, c_n)$

**Privacy-Preserving LDA.** To avoid releasing individual users' biometrics, we designed a three-party *privacy-preserving linear discriminant analysis* protocol, illustrated in Figure 2. The protocol is executed when new users enroll. The other two parties involved are the *enrollment server* (ES) and the *matrix publisher* (MP). The user generates biometric samples, encrypts them under the MP's public key and sends them to ES. ES stores the user's samples in encrypted form and computes, in conjunction with the user, the updated *encrypted* scatter within and scatter between matrices. The matrices are then sent to MP, which decrypts them and publishes the corresponding LDA matrix. We assume that MP does not collude with either ES or any user, as MP can decrypt any encrypted message. Interaction between the user and ES/MP is necessary only during enrollment. After that, the user is able to generate biometric keys using local data.

When using LDA for fuzzy commitments, the transformation matrix is also necessary in order to recover the key (i.e., to decommit). LDA matrix and scaling factors are not user-specific, therefore they only reveal information about the overall population. However, to take full advantage of the population data – especially to minimize FAR – users should update their LDA matrix periodically to include data from new users. (Update interval depends on a number of factors, such as the number of users, how many users join the system in a given time interval, etc.) After the matrix is updated and published, biometric keys must be re-generated since a biometric signal used with a different transformation matrix cannot not be used to reconstruct the key. During enrollment, each user $u$ holds a $m_u \times n$ matrix $X_u$ containing her $m_u$ training samples. The LDA algorithm creates a linear transformation that transforms the feature vectors to a space where the Fisher's Discriminant is maximal. Samples of all enrolled users are required to compute the transformation.

Our protocol guarantees that the ES does not learn any information about the users' input. Similarly, a new user do not learn information about the biometrics of existing users.

When a new user joins the system, an updated version of the LDA matrix is generated. In order to prevent the adversary from extracting information on the new user by comparing two consecutive LDA matrices, the ES provides its output to the MP only after a pre-determined number of users $w \gg 1$ have joined the system. This way, the adversary can only learn aggregate information of $w$ users.

## 4. Security Analysis

To show that our BKG technique is secure, we separately prove that it meets the BKG requirements from [2] – namely, that cryptographic keys are indistinguishable from random given the commitment (key randomness), and that given a cryptographic key and a commitment, no useful information about the biometric can be reconstructed (biometric privacy). Then, we show that the PP-LDA protocol is secure against a honest-but-curious adversary.

### 4.1. Key Randomness and Biometric Privacy

In order to define security of biometric key generation systems, Ballard *et al*. [2] introduced the notions of *Key Randomness* (REQ-KR), *Weak Biometric Privacy* (REQ-WBP) and *Strong Biometric Privacy* (REQ-SBP). We assume that the biometric is unpredictable after revealing $l_i$ least significant bits of each feature. Because the least significant bits of $x$ are the most affected by noise, we argue that these bits do not leak information about the $d - l_i$ most significant bits of each feature.

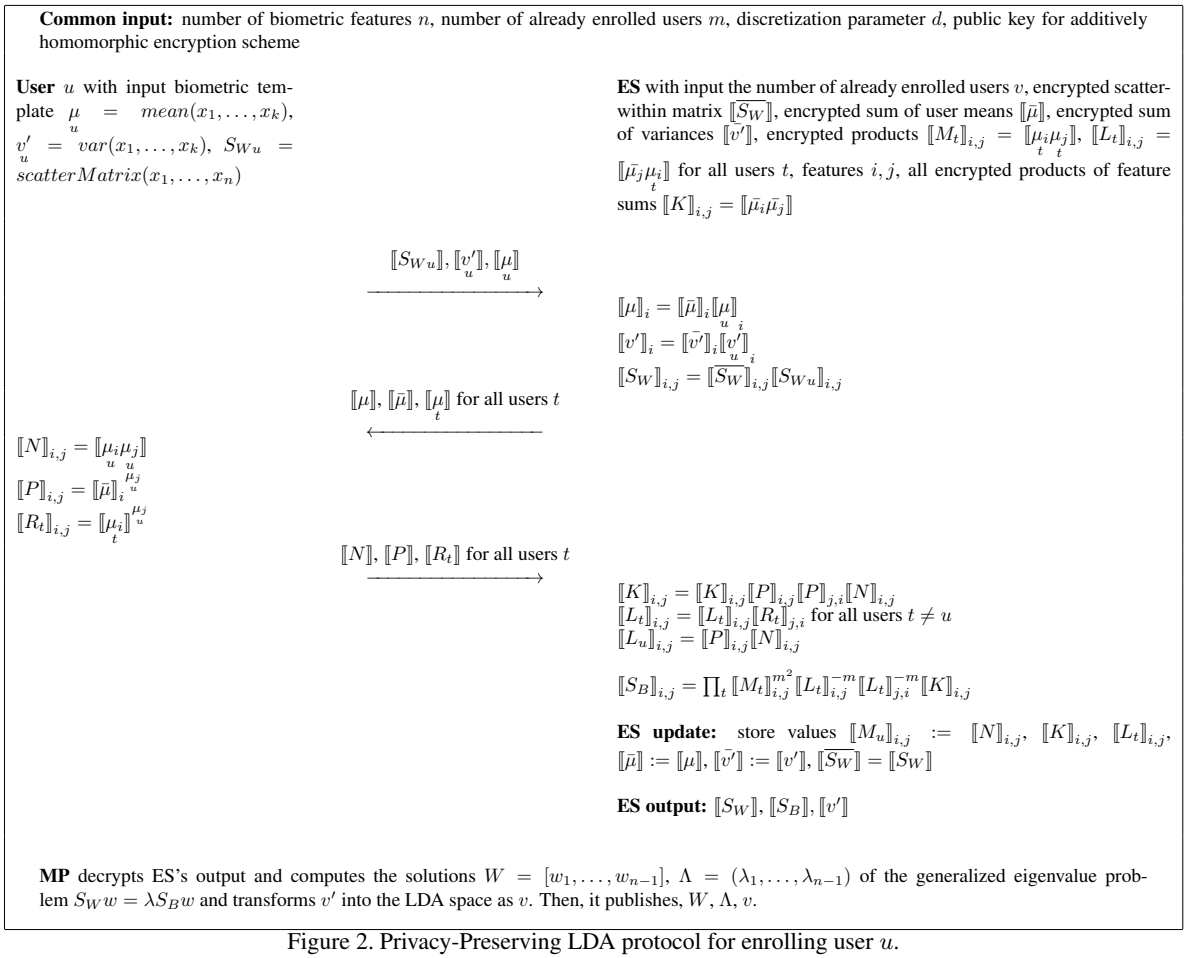


Figure 2. Privacy-Preserving LDA protocol for enrolling user $u$.

**Key Randomness** We formalize the notion of key randomness by defining Experiment IND-KR$_{\mathcal{A}}(\kappa)$:

**Experiment** IND-KR$_{\mathcal{A}}(\kappa)$

1. $\mathcal{A}$ is provided with a challenge $(\mathrm{PRF}_{c_i}(0), \delta)$, $k_b$ and $z$, where $k_0 = \mathrm{PRF}_{c_i}(z)$ and $k_1 \leftarrow_R \{0,1\}^\kappa$ for a bit $b \leftarrow_R \{0,1\}$, corresponding to user $i$.
2. $\mathcal{A}$ is allowed to obtain biometric information $x_j$ for arbitrary users $j$ such that $j \neq i$.
3. $\mathcal{A}$ outputs a bit $b'$ as its guess for $b$. The experiment outputs 1 if $b = b'$, and 0 otherwise.

**Definition 4.** *We say that a biometric key generation scheme has the Key Randomness property if there exist a negligible function* $\mathrm{negl}(\cdot)$ *such that for any PPT* $\mathcal{A}$, $\Pr[\text{IND-KR}_{\mathcal{A}}(\kappa) = 1] \leq 1/2 + \mathrm{negl}(\kappa)$.

**Theorem 2.** *Assuming that the* PRF *is a pseudo-random function family and that biometric* $X = (x_1, ..., x_n)$ *is unpredictable given* $l_i$ *least significant bits of each feature* $i$*, our Fuzzy Commitment scheme has the Key Randomness property.*

*Proof of Theorem 2 (Sketch).* Let $\mathcal{C}$ be a set of codewords such that $|\mathcal{C}| = 2^{\sum_{i=1}^{n}(d-l_i)}$ and the least significant $l_i$ bits of each symbol $i$ of all codewords in $\mathcal{C}$ are 0. If $c$ is selected uniformly from $\mathcal{C}$, the most significant $d - l_i$ bits in each codeword symbol $i$ are uniformly distributed in $\{0,1\}^{d-l_i}$. Since $x$ is unpredictable given the least significant $l_i$ bits in each feature and the most significant $d - l_i$ bits of each symbol $c_i$ are uniformly distributed, we have that $x$ is unpredictable given $\delta$. Because $c = x - \delta$, $c$ is unpredictable given $\delta$.

We now show that any PPT adversary $\mathcal{A}$ that has advantage $1/2 + \Delta(\kappa)$ to win the IND-KR$_{\mathcal{A}}(\kappa)$ experiment can be used to construct a distinguisher $\mathcal{D}$ that has similar advantage in distinguishing PRF from a family of uniformly distributed random functions.

$\mathcal{D}$ is given access to oracle $O(\cdot)$ that selects a random codeword $c$ and a random bit $b$, and responds to a query $q$ with random (consistent) values if $b = 1$, and with $\mathrm{PRF}_c(q)$ if $b = 0$. $\mathcal{D}$ selects a random $z$, $c' \leftarrow \mathcal{C}$ and $x' \leftarrow \mathcal{X}$, and sets $\delta' = x' - c'$. Then $\mathcal{D}$ sends $(O(0), \delta'), O(z)$ to $\mathcal{A}$.

$c$ is unpredictable given the least significant $l_i$ bits of each codeword symbol, and $\delta'$ is chosen from the same dis-

tribution as $\delta$. If $b = 0$, $(\mathrm{PRF}_c(0), \delta), \mathrm{PRF}_c(z)$ is indistinguishable from $(\mathrm{PRF}_{c'}(0), \delta'), \mathrm{PRF}_{c'}(z)$, because the $\delta$ and $\delta'$ follow the same distribution, $c$ and $c'$ are unpredictable given $\delta$ and thus the output of both $\mathrm{PRF}_c(\cdot)$ and $\mathrm{PRF}_{c'}(\cdot)$ are indistinguishable from random. If $b = 1$, then $O(\cdot)$ is a random oracle, so $(O(0), \delta'), O(1)$ is indistinguishable from pair $(\mathrm{PRF}_c(0), \delta), \mathrm{PRF}_c(1)$. $\delta$ and $\delta'$ are chosen from the same distribution and $c$ is unpredictable given $\delta$, so $\mathrm{PRF}_c(\cdot)$ is indistinguishable from random.

Eventually, $\mathcal{A}$ outputs $b'$, and $D$ outputs $b'$ as its guess. It is easy to see that $\mathcal{D}$ wins iff $\mathcal{A}$ wins, so $\mathcal{D}$ is correct with probability $1/2 + \Delta(\kappa)$. Therefore, if $\Delta(\cdot)$ is non-negligible, $D$ can distinguish PRF from a random function with non-negligible advantage over 1/2. However, this violates the security of the PRF; hence, $\mathcal{A}$ cannot exist. □

***Weak* and *Strong* Biometric Privacy.** REQ-WBP states that the adversary learns no useful information about a biometric signal from the commitment and the auxiliary information, while REQ-SBP states that the adversary learns no useful information about the biometric given auxiliary information, the commitment and the key. It is easy to see that, in our BKG algorithms, strong biometric privacy implies weak biometric privacy: key $k$ is computed as $\mathrm{PRF}_c(z)$; since the adversary knows $\mathrm{PRF}_c(0)$ as part of the commitment, $\mathrm{PRF}_c(z)$ does not add useful information.

We assume that the adversary has access to all public information – i.e., the LDA matrix, the vector of aggregate variances in the LDA space and all system parameters – and user-specific information such as the commitment $(\mathrm{PRF}_c(0), \delta)$, a list of keys computed as $k_i = \mathrm{PRF}_c(z_i)$ and a list of values $z_i$.

Since the output of $\mathrm{PRF}_c$ does not reveal $c$, $\mathrm{PRF}_c(0)$ and $k_i$-s do not disclose information about $x$. On the other hand, $\delta$ reveals the least significant $l_i$ bits of $x$. However, $x$ is unpredictable given its $l_i$ lest significant bits. Therefore, $x$ cannot be reconstructed from $\mathrm{PRF}_c(0)$, $k_i$ and $\delta$. Since $c$ is uniformly distributed in $\mathcal{C}$, $\delta$ does not reveal information about the most significant $d - l_i$ bits of $x$. Moreover, the $l_i$ least significant bits of $x$ are highly perturbed by noise and therefore do not reveal useful information about the biometric signal.

### 4.2. Security of LDA Protocol

We argue that the protocol in Figure 2 is secure, i.e., that ES does not learn any information about a specific user biometric, and that MP only learns $S_B$ and $S_W$. In particular, the user does not possesses the decryption key for the homomorphic encryption, and all messages from ES are encrypted. Since the encryption scheme is semantically secure, the user cannot extract any information from the protocol execution.

When interacting with the user, ES's view of the protocol consists in the encrypted values from the user, encrypted values from previous runs of the protocol and the number of users $v$. The output of the server is $[\![S_W]\!]$, $[\![S_B]\!]$ and $[\![v']\!]$. Because of the semantic security of the encryption scheme, ES cannot tell if the latter three values are replaced with encryptions of random elements. Therefore, the protocol does not reveal any information to ES.

During its interaction with MP, ES does not learn any additional information, because it does not receive any message from MP. MP receives encrypted values $[\![S_W]\!]$, $[\![S_B]\!]$, $[\![v']\!]$, that is able to decrypt. As we argue next, $S_W$, $S_B$ and $v$ do not leak information about a specific user if computer over a *set* of users.

### 4.3. Information Leakage through $S_W$ and $S_B$

Individual $S_{Wu}$ reveal significant information about a single user's biometric. For example, they leak feature variance and correlation between features for $u$. However, by averaging all users' *scatter within* matrices into $S_W$, information about single users is no longer available. In particular, the larger the number of users, and more uniform their selection, the closer $S_W$ will be to the value corresponding to the general population, which is assumed to be known. The same argument applies to both $S_B$ and $[\![\sigma']\!]$.

However, two tuples of elements $({S_W}^{t_1}, {S_B}^{t_1}, {\sigma'}^{t_1})$ and $({S_W}^{t_2}, {S_B}^{t_2}, {\sigma'}^{t_2})$ generated at different points in time $t_1, t_2$ reveal aggregate information corresponding to the users who enrolled between $t_1$ and $t_2$. If only a single user enrolls between $t_1$ and $t_2$, then it is possible to reconstruct $S_{Wu}$ as ${S_W}^{t_2} - {S_W}^{t_1}$. Therefore, in order to prevent this attack, $S_W$, $S_B$ and $v'$ should be updated in batches.

## 5. Experimental Evaluation

In order to quantify the biometric performance and key reconstruction reliability of our BKG technique, we performed free-text typing experiments on 486 volunteer subjects. Each subject was asked to answer between 10 and 13 questions, typing *at least* 300 character in each answer. Data was collected in two separate 45-to-120 minute sessions, which took place on different days. Experiments were performed using a custom Java GUI, which recorded keystroke with a 15.625 ms resolution, and on a standard QWERTY keyboard.

We used two feature subsets of features: the 23 most available *keyhold features*,[1] and keyhold features supplemented with 9 most available *digraph features*.[2] These features were chosen based on the availability in the first session. We then removed outliers by deleting all feature values higher than 500 ms. Finally, we discretized each feature in the range from 0 to 500 ms.

[1] 'Spacebar', 'E', 'O', 'I', 'A', 'S', 'H', 'N', 'R', 'T', 'L', 'D', 'U', 'Y', 'W', 'G', 'P', 'C', 'M', 'B', 'F', 'V', 'K'.

[2] 'HE', 'IN', 'TH', 'ER', 'AN', 'RE', 'EN', 'ND', 'HA'.

| | | keyhold+digraph | | | | keyhold only | | | |
|---|---|---|---|---|---|---|---|---|---|
| | minutes | entropy | FAR | FRR | availability | entropy | FAR | FRR | availability |
| with LDA | 4 | 99.95% | 5.6% | 6.8% | 81.5% | 97.7% | 6.9% | 14.1% | 94.4% |
| | 8 | 99.95% | 5.5% | 3.6% | 98.3% | 96.8% | 7.7% | 8.0% | 99.6% |
| w/o LDA | 4 | 81.9% | 9.2% | 9.8% | 82.7% | 62.1% | 12.7% | 15.1% | 94.4% |
| | 8 | 87.6% | 6.6% | 6.9% | 99.0% | 67.7% | 11.3% | 10.1% | 99.7% |

Table 1. BKG results. Whole training session was used for creating commitments, 4-minute and 8-minute slices from testing session were used for key retrieval. We report availability as the percentage of time slices that contain at least two vectors with all required features.

| | | computation | | | communication | |
|---|---|---|---|---|---|---|
| features | users | user | ES | MP | user-ES | ES-MP |
| 23 | 250 | 13 min 39 s | 40 min 43 s | 4 s | 17 MB | 135 KB |
| 23 | 500 | 27 min 9 s | 81 min 14 s | 4 s | 34 MB | 135 KB |
| 31 | 250 | 26 min 22 s | 78 min 42 s | 8 s | 33 MB | 260 KB |
| 31 | 500 | 52 min 33 s | 157 min 16 s | 8 s | 65 MB | 260 KB |

Table 2. Computational and communication overhead for PP-LDA protocol.

Data from the first session was used to create the commitments (biometric keys). For each user, we obtained per-feature variance and mean from the whole session. We used the mean to commit to the user's cryptographic key (see Section 3), and the per-user variance to compute the global variance.

### 5.1. False Accept/Reject and Availability

A standard metric for evaluating biometric systems is *equal error rate* (EER), which is defined as the value that FAR and FRR assume when they are equal. In our scenario, we have a false reject when a user's biometric fails to decommit the user's cryptographic material – i.e., when the biometric sample is not *close enough* to the original biometric. False accept is defined as the event when a user's biometric can be used to successfully decommit another user's cryptographic information.

When dealing with discrete systems, FAR and FRR might never assume the same value. In this case, we approximated EER by reporting both FAR and FRR at their minimum distance. Entropy is reported as the percentage of maximum Shannon entropy of discretized user templates in our dataset available through our BKG algorithm.

To evaluate FAR without LDA, we implemented a *zero-effort* impostor attack. This attack consists in employing a user's biometric to decommit other users' cryptographic keys. With LDA we used *cross-validation* instead of zero-effort attacks. We enrolled all users except for one, which we refer to as *impostor*. We then used impostor biometric data to attempt to decommit enrolled users' biometric keys. We repeated this experiments for each user, so that all of them could act as impostor. Using impostors that were not enrolled in the system gives better chance to succeeding in the attack, as the LDA transformation can maximize separation among users that are enrolled in the system.

Results are presented in Table 1 for both keyhold features alone and for keyhold with digraph. The results clearly show that using LDA improves both FAR/FRR and entropy. With 4-minute slices of testing data and using both keyhold and digraph features, LDA improved FAR-FRR from 9.2-9.8% to 5.6-6.8%. With 8-minute slices, the FAR-FRR improved from 6.6-6.9% to 5.5-3.6% using LDA.

One important issue to address is availability of the biometric. Each feature used for generating the key must also be used for decommitting, as both LDA and our error-correcting code cannot handle erasures. Our results show that 4-minute (8-minute) sample of keystrokes carries all the required information with probability greater than 94% (99%, respectively) for keystroke only, and over 81% (98%, respectively) samples have all the required keystrokes and digraphs. The FAR/FRR results are improved when more keystrokes/digraphs are available for each feature, at the cost of lower availability. Both results for 1 minimum sample and 2 minimum samples are provided in Table 1.

### 5.2. Computational Overhead of PP-LDA

The overhead in our privacy-preserving protocol is dominated by encryptions, decryptions and operations in the encrypted domain. We instantiated our protocol using the DKG [6, 5] cryptosystem with 1024bit key, 160bit subgroup size and 65bit plaintext size. We run our Java single-threaded prototype implementation on a desktop computer with Intel Xeon E5606 CPU at 2.13 GHz with 48 GB RAM running on Windows 7.

The amount of computation and communication depends on the number of features, indicated with $n$, and the number of enrolled users $m$. During the protocol, a new user performs $O(n^2)$ encryptions and $O(mn^2)$ exponentiations. The enrollment server computes $O(mn^2)$ multiplications and exponentiations, and the matrix publisher does $O(n^2)$ decryptions. The overall amount of communication in both rounds is $O(mn^2)$ between the user and enrollment server and $O(n^2)$ between the ES and MP.

The overhead of PP-LDA is reported in Table 2. Because

both the computation and communication depend on multiple parameters, we report representative parameter combinations corresponding to our settings.

Our experiments confirm that the cost of the PP-LDA protocol is relatively small, since the protocol is only executed once for each new user.

## 6. Related Work

**BKG based on Behavioral Biometrics.** Monrose *et al*. [13] evaluate the performance of BKG based on spoken password using data from 50 users. They report a false-negative rate of 4%.

Handwritten signature is another behavioral modality, where biometric key generation has been studied. Multiple papers, for example by Freire *et al*. [9], Feng *et al*. [7] and more recently Scheuermann *et al*. [19] evaluate the performance. The dataset sizes for the first three papers are 330, 25 and 144 users; the last paper does not include the number of users. The FRR/FAR presented in [9] are 57% and 1.7% respectively, with skilled forgeries, 8% EER in [7].

**BKG based on Physical Biometrics** Physical biometrics have also been used for biometric key generation, evaluated on fingerprints by Clancy [4] *et al*., Uludag *et al*. [21], Sy and Krishnan [20] and others. BKG on iris was studied by Rathgeb and Uhl [17], [16] and Wu *et al*. [22], and on face images by Chen *et al*. [3].

**Passwords Hardening** In [14], Monrose *et al*. use keystroke timing for increasing entropy (or "hardening") users passwords. Their technique extracts entropy from keystroke data, but does not use free-text and also does not generate keys.

**Privacy-Preserving LDA.** The problem of computing LDA on horizontally and vertically partitioned data has been addressed in [10] by Han and Ng. However, their protocol is not suitable in our setting. In particular, their technique addresses the problem where two parties have different partitions of a dataset, and want to compute a joint LDA matrix. In our scenario, however, we have many parties (the users) who want to compute a common LDA matrix.

## 7. Conclusion

Biometric key generation is an important and general primitive that can be used – among other things – for authentication, encryption and access control. In this paper we present the first BKG algorithm suitable for continuous authentication, based on keystroke dynamics. Our algorithm uses LDA to improve reliability and biometric performance. We therefore designed and implemented a secure privacy-preserving protocol for computing and updating LDA parameters using all users' biometric signals.

Biometric performance and computational overhead of our techniques are evaluated on a prototype implementation. Our experiments show that our BKG technique has low EER (between 3.6% and 5.5%), and limited overhead.

## References


[1] Anonymized.

[2] L. Ballard, S. Kamara, and M. Reiter. The practical subtleties of biometric key generation. In *USENIX Security Symposium*, 2008.

[3] B. Chen and V. Chandran. Biometric based cryptographic key generation from faces. In *Digital Image Computing Techniques and Applications*, 2007.

[4] T. Clancy, N. Kiyavash, and D. Lin. Secure smartcardbased fingerprint authentication. In *ACM SIGMM Workshop on Biometrics Methods and Applications*, 2003.

[5] I. Damgård, M. Geisler, and M. Krøigård. A correction to efficient and secure comparison for on-line auctions. Cryptology ePrint Archive, Report 2008/321, 2008.

[6] I. Damgård, M. Geisler, and M. Krøigård. Homomorphic encryption and secure comparison. *IJACT*, 2008.

[7] H. Feng and C. Wah. Private key generation from on-line handwritten signatures. *Information Management & Computer Security*, 10(4), 2002.

[8] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7), 1936.

[9] M. Freire-Santos, J. Fierrez-Aguilar, and J. Ortega-Garcia. Cryptographic key generation using handwritten signature. In *Defense and Security Symposium*, 2006.

[10] S. Han and W. Ng. Privacy-preserving linear fisher discriminant analysis. In *PAKDD*, 2008.

[11] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *CCS*, 1999.

[12] C. Lee. Some properties of nonbinary error-correcting codes. *IRE Transactions on Information Theory*, 4(2), 1958.

[13] F. Monrose, M. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. In *S&P*, 2001.

[14] F. Monrose, M. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *Int. J. Inf. Sec.*, 1(2), 2002.

[15] M. Naor and O. Reingold. From unpredictability to indistinguishability: A simple construction of pseudo-random functions from MACs. In *CRYPTO*, 1998.

[16] C. Rathgeb and A. Uhl. An iris-based interval-mapping scheme for biometric key generation. In *ISPA*, 2009.

[17] C. Rathgeb and A. Uhl. Privacy preserving key generation for iris biometrics. In *Communications and Multimedia Security*, 2010.

[18] R. Roth. *Introduction to Coding Theory*. Cambridge University Press, New York, NY, USA, 2006.

[19] D. Scheuermann, B. Wolfgruber, and O. Henniger. On biometric key generation from handwritten signatures. In *BIOSIG*, 2011.

[20] B. Sy and A. Krishnan. Generation of cryptographic keys from personal biometrics: An illustration based on fingerprints. In J. Yang and S. Xie, editors, *New Trends and Developments in Biometrics*. InTech, 2012.

[21] U. Uludag, S. Pankanti, and A. Jain. Fuzzy vault for fingerprints. In *AVBPA*, 2005.

[22] X. Wu, N. Qi, K. Wang, and D. Zhang. A novel cryptosystem based on iris key generation. In *International Conference on Natural Computation*, 2008.

[23] Y. Wu and C. Hadjicostis. Decoding algorithm and architecture for BCH codes under the Lee metric. *Transactions on Communications*, 56(12), 2008.